

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 02-217925  
(43)Date of publication of application : 30.08.1990

---

(51)Int.CI. G06F 9/42  
G06F 9/46

---

(21)Application number : 01-037805 (71)Applicant : TOSHIBA CORP  
TOSHIBA MICRO ELECTRON KK  
(22)Date of filing : 17.02.1989 (72)Inventor : SAKAGAMI KENJI

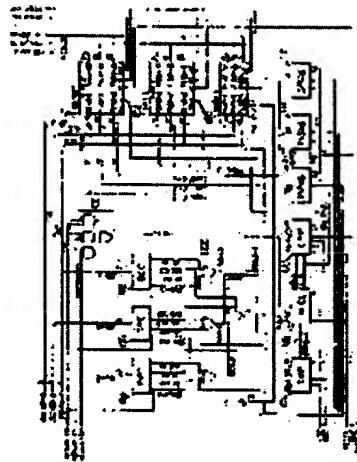
---

**(54) MICROPROCESSOR**

**(57)Abstract:**

**PURPOSE:** To improve cost performance by flexibly determining the number of registers to be allocated to windows in accordance with the number of registers to be used for a procedure in a program

**CONSTITUTION:** A microprocessor is constituted of R-FILE1 modules 101, 102, an R-FILE2 module 103, decoders(DECs) 104 to 106, a current window pointer (CWP) 107, a register(W-CC) 108, a current block pointer (CBP) 109, and substructors(INSAGs) 110 to 112. A working register to be used for each procedure is determined in accordance with the individual procedure in the program and the number of registers constituting the working register is determined. Consequently, the redundancy of the system is removed and the cost performance is improved.



---

**LEGAL STATUS**

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

BEST AVAILABLE COPY

## ⑫公開特許公報(A) 平2-217925

⑬Int.Cl.<sup>3</sup>G 06 F 9/42  
9/46識別記号 330 R  
313 C府内整理番号 7361-5B  
8945-5B

⑭公開 平成2年(1990)8月30日

審査請求 未請求 請求項の数 2 (全12頁)

⑮発明の名称 マイクロプロセッサ

⑯特 願 平1-37805

⑯出 願 平1(1989)2月17日

⑰発明者 坂上 健二 神奈川県川崎市川崎区駅前本町25番地1 東芝マイコンエンジニアリング株式会社内

⑯出願人 株式会社東芝 神奈川県川崎市幸区堀川町72番地

⑯出願人 東芝マイクロエレクトロニクス株式会社 神奈川県川崎市川崎区駅前本町25番地1

⑯代理人 弁理士 佐藤 一雄 外3名

## 明細書

## 1. 発明の名称

マイクロプロセッサ

前記記憶手段中の前記割り当て数に基づいて、前記各レジスタウィンドウが前記レジスタのうちのどれによって割り当てるかを決定する第2決定手段と、を備えることを特徴とするマイクロプロセッサ。

## 2. 特許請求の範囲

1. 複数個のレジスタウィンドウを有し、前記各レジスタウィンドウは複数個のレジスタを有するマイクロプロセッサであって、

前記各レジスタは前記レジスタウィンドウのうちの複数のものに対して割り当てられており、さらには、

プログラム上の個々のプロシージャに対して前記レジスタウィンドウのうちのいずれかを割り当てるにより、前記各プロシージャ上で使用されるワーキングレジスタを決定する第1決定手段と、

前記各レジスタウィンドウへのレジスタ割り当て数を、前記ワーキングレジスタに対するアドレス情報に基づいて自動的に記憶する記憶手段と、

複数個のレジスタウィンドウを有し、前記各レジスタウィンドウは複数個のレジスタを有し、さらにCPUを有するマイクロプロセッサであって、

前記各レジスタは前記レジスタウィンドウのうちの複数のものに対して割り当てられており、さらには、

プログラム上の個々のプロシージャに対して前記レジスタウィンドウのうちのいずれかを割り当てるにより、前記各プロシージャ上で使用されるワーキングレジスタを決定する第1決定手段と、

前記各レジスタウィンドウへのレジスタ割り当て数を、前記CPUが実行できる命令の1つに基づいて記憶する記憶手段と、

## 特開平2-217925 (2)

前記記憶手段中の前記割り当て数に基づいて、前記各レジスタウインドウが前記レジスタのうちのどれによって構成されるかを決定する第2決定手段と、を備えることを特徴とするマイクロプロセッサ。

### 3. 発明の詳細な説明

#### (発明の目的)

#### (産業上の利用分野)

本発明は、マイクロプロセッサに関し、特に、RISC (Reduced Instruction Computer Set) タイプのマイクロプロセッサに関する。

#### (従来の技術)

SPARCアーキテクチャーによるマイクロプロセッサは、例えば、刊行物『10 MIPS から 100 MIPS までを単一アーキテクチャーでカバーする SPARC』(NIKKEI BYTE/MARCH 1988) に示されている。ここには、レジスタウインドウを用いたマイクロプロセッサに関して詳述されている。このようなシステムで使用する、プロシ

ャコールを多用するプログラムにおいては、多くのウインドウを使用することになる。

従って、マイクロプロセッサチップ上のウインドウ数が少ないとウインドウのオーバーフロー／アンダーフロートラップ処理によるオーバーヘッドが大きな問題となってくる。SPARCアーキテクチャーでは、1つのウインドウを追加するためには、32ビット長のレジスタを16本必要とする。しかしながら、チップ上のウインドウ数はハードウェア盤によって制限を受けるため、現実には多くすることはできない。ちなみに、SPARCでは、ウインドウの数は最小6個、最大32個と規定しているが、1.5  $\mu$ mC-MOSゲートアレイによるチップの数は、上記刊行物では7個となっている。

#### (発明が解決しようとする課題)

従来のマイクロプロセッサは以上のように構成されているので、1個のウインドウに割り当てるレジスタ数は固定である(以下これを固定方式と呼ぶ)。しかしながら、個々のプロシージャコー

ルで実際に使用するレジスタ数は割り当て数より少ないと場合も多くあり、割り当て数が固定であるということは非常に冗長である。このため、例えば、現在のウインドウで未使用のレジスタを別のウインドウに割り当てるウインドウ数を増やすといった、レジスタの有効活用ができない。

このような問題を解決する目的で、1つのウインドウに割り当てるレジスタ数を可変にする方式が従来技術として知られている(以下、この方式を可変方式と呼ぶ)。

本発明は、可変方式のレジスタウインドウを含むマイクロプロセッサの構成技術に関し、その目的の1つは、プロセッサの実行プログラムの内容に応じて、自動的に1つのウインドウに割り当てるレジスタ数を決定する可変方式のレジスタウインドウの構成技術を提供し、固定方式のレジスタウインドウによるマイクロプロセッサの為に開発されたプログラムをそのまま実行できる可変方式のレジスタウインドウによるプロセッサの構成技術を提供することにより、上記プログラムの内容

の変更なしに従来のプログラムの処理速度を、ウインドウのオーバーフロー／アンダーフローのオーバーヘッドを低減することにより、改善出来るようにすることにある。

目的のもうひとつは、可変方式のレジスタウインドウによるマイクロプロセッサで、その命令セットの中に、ウインドウへのレジスト割り当て数を制御する命令を設けたマイクロプロセッサ技術を提供し、プログラム内のプロシージャの使用するレジスタ数の規模に応じて柔軟にウインドウへのレジスタ割り当て数を決定できるようにし、使い勝手が良く、かつウインドウの使用効率を高め、ウインドウのオーバーフロー／アンダーフローによるオーバーヘッドを低減でき、コストパフォーマンスの優れたマイクロプロセッサを提供することにある。

#### (発明の構成)

#### (課題を解決するための手段)

本発明の第1のマイクロプロセッサは、複数個のレジスタウインドウを有し、前記各レジスタウ

インドウは複数個のレジスタを有するマイクロプロセッサであって、

前記各レジスタは前記レジスタウインドウのうちの複数のものに対して割り当てられており、さらに、

プログラム上の個々のプロシージャに対して前記レジスタウインドウのうちのいずれかを割り当てるにより、前記各プロシージャ上で使用されるワーキングレジスタを決定する第1決定手段と、

前記各レジスタウインドウへのレジスタ割り当て数を、前記ワーキングレジスタに対するアドレス情報に基づいて自動的に記憶する記憶手段と、

前記記憶手段中の前記割り当て数に基づいて、前記各レジスタウインドウが前記レジスタのうちのどれによって構成されるかを決定する第2決定手段と、を備えるものとして構成される。

本発明の第2のマイクロプロセッサは、複数個のレジスタウインドウを有し、前記各レジスタウインドウは複数個のレジスタを有し、さらにCPU

を有するマイクロプロセッサであって、

前記各レジスタは前記レジスタウインドウのうちの複数のものに対して割り当てられており、さらに、

プログラム上の個々のプロシージャに対して前記レジスタウインドウのうちのいずれかを割り当てるにより、前記各プロシージャ上で使用されるワーキングレジスタを決定する第1決定手段と、

前記各レジスタウインドウへのレジスタ割り当て数を、前記CPUが実行できる命令の1つに基づいて記憶する記憶手段と、

前記記憶手段中の前記割り当て数に基づいて、前記各レジスタウインドウが前記レジスタのうちのどれによって構成されるかを決定する第2決定手段と、を備えるものとして構成される。

#### (作用)

各レジスタウインドウ(モジュール)へのレジスタ割り当て数が記憶手段に記憶されている。第2決定手段は、その割り当て数に応じて、各モジ

ュールがどのレジスタによって、即ち何個のレジスタによって構成されるかを決定する。従って、記憶手段中の割り当て数によって、各モジュールが何個のレジスタによって構成されるかが変化することとなる。即ち、プログラム上の個々のプロシージャに応じて、そのプロシージャ上で使用されるワーキングレジスタが第1決定手段で決定されると共にワーキングレジスタを構成するレジスタの数が第2決定手段で決定される。

さらに、第1の発明においては、記憶手段での、各レジスタウインドウへのレジスタ割り当て数の記憶は、ワーキングレジスタに対するアドレス情報に基づいて自動的に行われる。また、第2の発明においては、前記割り当て数の記憶は、CPUの実行できる命令の1つに基づいて行われる。

#### (実施例)

以下、図面を参照しながら本発明の実施例を説明する。

第1図は本発明の一実施例に係るマイクロプロセッサのブロック図で、特にウインドウの

locals、outsに割り当てるレジスタ数を8本または4本とすることを可能とした例を示すものである。

第1図において、レジスタアドレスrs1-A0~4はソースレジスタ1を、レジスタアドレスrs2-A0~4はソースレジスタ2を、レジスタアドレスrd-A0~4はデステネーションレジスタをそれぞれ指定するものである。前記各アドレスのうちのA0~2はそれぞれR-FILE1モジュール101、R-FILE1モジュール102、R-FILE2モジュール103に与えられる。より詳しくは、アドレスrs1-A0~2は各モジュールのポートP0-A0~6に、アドレスrs2-A0~2はポートP1-A0~6に、アドレスrd-A0~4はポートP2-A0~6にそれぞれ与えられる。また、各アドレスA2は論理ブロック100に与えられる。そして、各アドレスのA3~4はそれぞれデコーダ(DEC)104、105、106に与えられる。一方、rs1-D0~31はソースレジスタ1

の出力データであり、 $r_s 2-D0 \sim 31$  はソースレジスタ2の出力データである。これらの $r_s 1-D0 \sim 31$  および $r_s 2-D0 \sim 31$  のいずれも R-FILE1モジュール101、R-FILE1モジュール102、R-FILE2モジュール103のそれぞれのポート P0-D0～31、P1-D0～31 から出力される。また、 $r_d-D0 \sim 31$  はデステネーションレジスタへの入力データであり、R-FILE1モジュール101、R-FILE1モジュール102 および R-FILE2モジュール103 の各ポート P2-D0～31 に与えられる。

なお、A0～4で示されるアドレスは globals 8本と1個のウインドウ24本 (ins 8本、locals 8本、outs 8本) の計32本のワーキングレジスタの指定を行なうものである。D0～31 で示されるデータはアドレス A0～4 に対応するワーキングレジスタの入出力データである。

なお、プログラム上の基本演算は2つのソースレジスタと1つのデステネーションレジスタとの

間で見かけ上1サイクルで行なうようにされている。ワーキングレジスタを構成する R-FILE1モジュール101、R-FILE1モジュール102 および R-FILE2モジュール103 は、データ出力ポートが2個、データ入力ポートが1個の3ポート構成となっている。各アドレスおよびデータは、R-FILE1モジュール101、R-FILE1モジュール102 および R-FILE2モジュール103 の各ポートに対して独立に接続されている。

R-FILE1モジュール101 は ins/outs のワーキングレジスタを構成するレジスタファイルのモジュールで、32ビット×56本の構成を有する。R-FILE1モジュール102 は locals のワーキングレジスタを構成するレジスタファイルのモジュールで、32ビット×56本の構成を有する。また、R-FILE2モジュール103 は globals のワーキングレジスタを構成するレジスタファイルのモジュールで、32ビット×8本の構成を有する。

ちなみに、各 R-FILE1モジュール101、R-FILE1モジュール102 および R-FILE2モジュール103において、 $\ominus$  はクロック CLK の入力端子、RE はリードイネーブル REGR-EN の入力端子、WE はライトイネーブル REGW-EN の入力端子、S0～2 は各ポートの選択を行なう信号の入力端子である。

R-FILE1モジュール101 の内部は、第2図に示すように、0～13の番地付けのなされた14個のブロックに分割されている。さらに、1つのブロックは4本のレジスタにより構成される。そして、ポート P0 のアドレス A3～6 (上位4ビット) がブロックアドレスになり、ポート P0 のアドレス A0～2 (下位3ビット) がポート P0 のアドレス A3～6 が指定するブロックをベースとした8本のレジスタに対するアドレスになる。ポート P1 のアドレス A0～6、ポート P2 のアドレス A0～6 も上記と同様である。ウインドウ13が8本モードであるときのレジスタファイルの割り当てブロックは、locals は R-

FILE1モジュール102 のブロック13、12、outs は R-FILE1モジュール101 のブロック13、12、ins は R-FILE1モジュール101 のブロック0 (初期状態) である。また、ウインドウ13が4本モードであるときのレジスタファイルの割り当てブロックは locals は R-FILE1モジュール102 のブロック13、outs は R-FILE1モジュール101 のブロック13、ins は R-FILE1モジュール101 のブロック0 (初期状態) となる。この他、S0～2 の入力信号によってポート P0～2 がイネーブルかそうでないかを決定する。ちなみに、ポート P0～2 がイネーブルとなるのは S0～2 の入力信号が “1” の時である。

また、カレントウインドウポインタ (CWP) 107 は現在のプロシージャが使用しているウインドウの番号を出力するものであり、且つインクリメント信号 INC に基づいてインクリメント可能な1ビットのバイナリカウンタである。ちなみに、SPARC ではカレントウインドウポインタ

(CWP) はプロセッサステートレジスタ (PSR) の下位5ビットを形成しているので、本実施例ではこれに準じて5ビットとしている。このカレントウインドウポインタ (CWP) 107は、リセット信号RESETが入力されると、カウント値がプロシージャが最初に使用するウインドウの番号CWP-13にセットされる。ちなみに、SPARCではウインドウ番号の最も大きいものから使用していくので、本実施例はこれに準じている。なお、CWP-0の状態でディクリメント信号DECによりディクリメントするとCWP-13に戻る。そして、カレントウインドウポインタ (CWP) 107の出力信号CWP0~4は次に説明するレジスタ (W-CC) 108のビットを指定する。

レジスタ (W-CC) 108は、各ウインドウが、8本モードかあるいは4本モードかの情報を記憶する14ビットのレジスタである。14ビットはそれぞれ14個のウインドウに対応している。各ビットは“0”が4本モードに、“1”が8本

モードにそれぞれ対応している。これは、RESETが入力されると全ビットが“0”にリセットされ、4本モードに設定される。8本モードへの切り替えは、次のようにして自動的に実施される。即ち、outsとlocalsのワーキングレジスタのそれぞれの8個のアドレスの組に前半と後半の4個に分ける。プロシージャでouts、localsとも後半の4個しかアクセスせずに次のウインドウに移った場合は、そのウインドウは4本モードのままになっている。もし、outs、localsのいずれかで、前半のいずれかのレジスタをアクセスすると、レジスタ (W-CC) 108の対応するビットが“1”にセットされ、このウインドウは8本モードとなる。localsの前半はr16~r19、後半はr20~r23、outsの前半はr8~r11、後半はr12~r15である。

一方、許可信号r1CCW-ENは、ソースレジスタ1のレジスタアドレスのr1-A2を、レジスタ (W-CC) 108のセットに使用することを許可するものである。また、許可信号r1

2CCW-ENは、ソースレジスタ2のレジスタアドレスのr2-A2を、レジスタ (W-CC) 108のセットに使用することを許可するものである。そして、許可信号rdCCW-ENは、デスクネーションレジスタのレジスタアドレスのrd-A2を、レジスタ (W-CC) 108のセットに使用することを許可するものである。これらの許可信号r1CCW-EN、r2CCW-EN、rdCCW-ENは各アドレスA2が入力される論理ブロック100に入力され、結果としてレジスタ (W-CC) 108の特定のビットを“1”にセットするセット信号SET-CCを出力する。ちなみに、この場合のビットの指定は、カレントウインドウポインタ (CWP) 107からのCWP0~4で行なうことになる。

論理ブロック100からのセット信号SET-CCとカレントウインドウポインタ (CWP) 107からのCWP0~4とによってビット指定されたレジスタ (W-CC) 108は、ビット出力ccnとビット出力ccn-1をカレントブ

ックポインタ (CBP) 109に送出する。ちなみに、ビット出力ccnは現在のウインドウに対応するレジスタ (W-CC) 108のビット状態であり、ビット出力ccn-1は前のウインドウに対応するレジスタ (W-CC) 108のビット状態である。

カレントブロックポインタ (CBP) 109は現在のウインドウが実際に使用するR-FILE 1モジュール101でのブロックを指定する4ビットのバイナリカウンタであり、“-1”、“-2”、“+1”、“+2”的動作が可能である。このカレントブロックポインタ (CBP) 109は、リセット信号RESETが入力されると、プロシージャが最初に使用するウインドウに対してR-FILE 1モジュール101のブロック13を割り当てる。ちなみに、CWP-0の状態でディクリメント信号DECによりディクリメントするとCWP-13に戻る。このカレントブロックポインタ (CBP) 109の出力CWP-CBP0~3に

## 特開平2-217925 (6)

よって、ウインドウに割り当てるレジスタ数が制御される。なお、この出力 C B P - C b P 0 ~ 3 はウインドウのオーバーフロー／アンダーフローの情報を外部に知らせる機能をも有する。そして、CWPがCWP-1となって次のウインドウに移るとき、ビット出力 c c n が“1”ならCBPがCBP-2となり、現在のウインドウに対して8本のレジスタ（2個のブロック）を割り当てる。一方、ビット出力 c c n = “0”ならばCBPがCBP-1となり現在のウインドウに対して4本のレジスタ（1個のブロック）を割り当てる。これに対して、CWPがCWP+1となり前のウインドウに戻るときはビット出力 c c n - 1 が“1”ならばCBPがCBP+2となり、ビット出力 c c n - 1 が“0”ならばCBPがCBP+1となる。

サブストラクタ (INSAG) 110, 111, 112は、レジスタアドレスが l o s を指定したとき、l o s / o u t s の R - FILE 1 モジュール 101 へのブロック指定アドレスを o u t s の値から

l o s の値に変換する4ビットのサブストラクタであり、R - FILE 1 モジュール 101 のポート P 0, P 1, P 2 にそれぞれ対応して設けられる。そして、l o s が指定されてビット出力 c c n - 1 が“1”的ときに、サブストラクタ (INSAG) 110, 111, 112 からの各出力信号 C b P 0' 0 ~ 3, C b P 1' 0 ~ 3, C b P 2' 0 ~ 3 の各値は C B P - 2 となり、ビット出力 c c n - 1 が“0”的時にサブストラクタ (INSAG) 110, 111, 112 からの出力信号 C b P 0' 0 ~ 3, C b P 1' 0 ~ 3, C b P 2' 0 ~ 3 の各値は C B P - 1 となる。一方、o u t s が指定されるとサブストラクタ (INSAG) 110, 111, 112 からの出力 C b P 0' 0 ~ 3, C b P 1' 0 ~ 3, C b P 2' 0 ~ 3 の各値としては C B P - 0 ~ 3 がそのままスルーリング出力される。

デコーダ (DEC) 104, 105, 106 はそれぞれレジスタアドレス r s 1 - A 0 ~ 4, r s 2 - A 0 ~ 4, r d - A 0 ~ 4 のうちの各上

位2ビット A 3 ~ 4 により、g l o b a l s 、l o s 、l o c a l s のうちのいずれかのレジスタアドレスになっているかをデコードするものである。そして、A 3 = “0”、A 4 = “0”的時には、デコーダ (DEC) 104 からは g l o b - r s 1 が、デコーダ (DEC) 105 からは g l o b - r s 2 が、デコーダ (DEC) 106 からは g l o b - r d がそれぞれ出力される。また、A 3 = “1”、A 4 = “0”的時には、デコーダ (DEC) 104 からは l o c - r s 1 が、デコーダ (DEC) 105 からは l o c - r s 2 が、デコーダ (DEC) 106 からは l o c - r d がそれぞれ出力される。また、A 3 = “0”、A 4 = “1”的時には、デコーダ (DEC) 104 からは l o c - r s 1 が、デコーダ (DEC) 105 からは l o c - r s 2 が、デコーダ (DEC) 106 からは l o c - r d がそれぞれ出力される。そして、A 3 = “1”、A 4 = “1”的時には、デコーダ (DEC) 104 からは o u t - r s 1 が、デコーダ (DEC) 105 からは o u t - r s 2 が、デコーダ (DEC) 106 からは o u t - r d がそれぞれ出力される。

これらの信号のうち l o c - r s 1 、l o c - r s 2 、l o c - r d と o u t - r s 1 、r s 2 、r d とは論理ブロック 200 で処理され、I O S E L 0 ~ 2 という R - FILE 1 モジュール 101 の各ポートをイネーブルにする信号として送出されて、R - FILE 1 モジュール 101 の入力端子 S 0 ~ 2 に入力される。また、l o c - r s 1 、l o c - r s 2 、l o c - r d は R - FILE 1 モジュール 102 の各ポートをイネーブルにする信号として送出され R - FILE 1 モジュール 102 の入力端子 S 0 ~ 2 に入力される。また、g l o b - r s 1 、g l o b - r s 2 、g l o b - r d は R - FILE 1 モジュール 103 の各ポートをイネーブルにする信号として送出され R - FILE 1 モジュール 103 の入力端子 S 0 ~ 2 に入力される。ちなみに、ワーキングレジスタのアドレスは r 0 ~ 7 が g l o b a l s に、r 8 ~ 15 が o u t s に、r 16 ~ 23 が l o c a l s に、r 24 ~ 31 が l o s にそれぞれ対応する。

以上のような構成において、次にその動作を第3図のタイミングチャートに基づいて説明する。

## 特開平2-217925 (7)

ちなみに、第3図 (A) はクロック CLK、(B) はディクリメント信号 DEC、(C) はインクリメント信号、(D) は R-FILE 1 モジュール 101、R-FILE 1 モジュール 102 および R-FILE 2 モジュール 103 に与えられるリードイネーブル REGR-EN、(E) は R-FILE 1 モジュール 101、R-FILE 1 モジュール 102 および R-FILE 2 モジュール 103 に与えられるライトイネーブル REGW-EN、(F) は各レジスタアドレスをレジスタ (W-CC) 108 にセットすることを許可する許可信号  $r_s 1\text{CCW-EN}$ 、 $r_s 2\text{CCW-EN}$ 、 $r_b\text{CCW-EN}$ 、(G) はレジスタアドレス  $r_s 1-A0\sim4$ 、(H) はレジスタアドレス  $r_s 2-A0\sim4$ 、(I) はレジスタアドレス  $r_d-A0\sim4$ 、(J) はソースレジスタ 1 の出力データ  $r_s 1-D0\sim31$ 、(K) はソースレジスタ 2 の出力データ  $r_s 2-D0\sim31$ 、(L) はデステネーションレジスタの出力データ  $r_d-D0\sim31$ 、(M) はカレントウインドウポイン

タ (CWP) 107 の出力 CWP0~4、(N) はレジスタ (W-CC) 108 のビット出力  $c_{cn}$ 、(O) はレジスタ (W-CC) 108 のビット出力  $c_{cn-1}$ 、(P) はカレントプロックポイント (CBP) 109 の出力である CBP0~3、(Q) はサブストラクタ (INSAG) 110 から送出される CBP0' 0~3、(R) はサブストラクタ (INSAC) 111 から送出される CBP1' 0~3、(S) はサブストラクタ (INSAG) 112 から送出される CBP2' 0~3、(T) はデコーダ (DEC) 104 から出力される glob-rs1、(U) デコーダ (DEC) 104 から出力される ls-rs1、(V) はデコーダ (DEC) 104 から出力される loc-rs1、(W) はデコーダ (DEC) 104 から出力される out-rs1、(X) は論理ブロック 100 から出力されるセット信号 SET-CC である。

第3図に示した命令のシーケンスは①で ADD、②で SAVE、③で ADD、④で SAVE となっ

ている。①ADD では  $rd$  の  $r_7$  に  $rs1$  の  $r_31$  と  $rs2$  の  $r_12$  を加算して移す。次に、②SAVE では次のウインドウに移り、CWP を CWP-1 とする。そして、③ADD では  $rd$  の  $r_6$  に  $rs1$  の  $r_16$  と  $rs2$  の  $r_12$  を加算して移す。そして、次の④SAVE でセーブする。ちなみに、この場合 [ $r_31$ ] には 7 が、 [ $r_12$ ] には 3 が、 [ $r_16$ ] には 2 がそれぞれ予め入っているものとする。文献（前記刊行物）による SPARC チップでは命令を 4 段のバイブラインで実行しているので、本実施例ではこれに準じて 4 段のバイブラインとする。次に、第4図に基づいてバイブルайн構造の概略を説明する。ここでは、ADD 命令を例にとって各ステージでの処理を説明する。

まず、ADD 命令をメモリ 200 から取り出し、インストラクションレジスタ (IR) 202 に書き込む (ステップ F)。次に、インストラクションレジスタ (IR) 202 の中の命中をインストラクションデコーダ (IDEC) 204 に入力してこれをデコードする。そして、次の命令に対す

るアドレスを、アドレスジェネレータ (AG) 206 およびプログラムカウンタ (PC) 208 により発生する。レジスタアドレス  $r_s 1$ 、 $r_s 2$  (2つのソースレジスタのデータ) をレジスタファイル (RF) 210 から読み出し、A レジスタ 212、B レジスタ 214 に書き込む (ステップ D)。

A レジスタ 212 のデータと B レジスタ 214 のデータを演算ユニット ALU 216 で加算して、結果を結果レジスタ (R) 218 に書き込む (ステップ E)。

結果レジスタ (R) 218 のデータを  $rd$  の示すデステネーションレジスタに書き込む (ステップ S)。

第3図のタイミングチャートでは上記各ステップをクロック CLK の  $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$  に対応づけて示している。即ち、まず、①の ADD においては、 $t_2$  のステップ D でレジスタアドレス  $r_s 1-A0\sim4\sim31$ 、レジスタアドレス  $r_s 2-A0\sim12$  となりレジスタファイルから

$r_{31}$ と $r_{12}$ のデータを読み出す。但し、実際の読み出し動作は $t_2$ の後半で行なっているので、データ $r_{s1-D0 \sim 31}$ 、データ $r_{s2-D0 \sim 31}$ にデータが出力されるのは $t_2$ の後半となる。 $r_{31}$ のデータは3であるので、データ $r_{s1-D0 \sim 31} = 7$ 、データ $r_{s2-D0 \sim 31} = 3$ となる。そして、 $t_3$ のステップEでは $r_{31} + r_{12} = 10$ を実行し、結果をレジスタに書き込む。 $r_{31}$ はinsレジスタなので、 $r_{31}$ の読み出しのときにはCBP-1で $CbP0' 0 \sim 3-0$ となる。次の、 $t_4$ のステップSでは、結果レジスタのデータをデステネーションレジスタの $r_7$ に書き込む。書き込みは $t_4$ の前半で行なう。なお、 $t_4$ の前半においては、②のADD命令のためにソースレジスタからの読み出しが行なわれている。

②のSAVEでは、 $t_4$ のステップEでCWP-CWP-1を実行する。①の実行において、localsへのアクセスがなく且つoutsへのアクセスにおいても $r_8 \sim r_{15}$ の中で後半の $r_{12}$ へのア

クセスであったので、レジスタ(W-CC)108のビット13 = "0" のままであるので、つまりウインドウ13は4本モードのままであるので、CBP-CBP-1を実行する。

そして、③のADDでは、 $t_4$ のステップDにおいて、 $r_{16}$ と $r_{12}$ のデータ2, 3がデータ $r_{s1-D0 \sim 31}$ 、 $r_{s2-D0 \sim 31}$ に出力される。このとき、 $r_{16}$ はlocalsの前半のレジスタなので、セット出力SET-CC = "1" となり、レジスタ(W-CC)108のビット12に"1"がセットされる。その結果、ウインドウ12は8本モードになる。

次の、④のSAVEでは、 $t_4$ のステップEで、CWP-CWP-1を実行する。ウインドウ12は8本モードになっているので、CBP-CBP-1を実行する。

以上の例では、localsとoutsの割り当て数をブロック単位で管理しており、localsとins/outsのレジスタ数が等しく、さらにlocalsとins/outsを分離して別のモジュールにしている。その結果、

ウインドウ1個当たりのコンディション情報が1ビットでよく、実アドレス発生回路が簡単になり、実アドレスの計算も非常に高速に実行できるようになるというメリットがある。

なお、上記実施例では8本と4本の切り替えを実示したが、レジスタの割り当て数は8本～0本の範囲で任意に選ぶことができる。また、localsとoutsへのレジスタの割り当て数を異なるようにしても良い。

ちなみに、ウインドウに対するレジスタ数を可変とすれば、レジスタ(W-CC)108としては14ビットレジスタが、カレントブロックポインタ(CBP)109としては4ビットカウンタが、およびサブストラクタ(INSAG)110, 111, 112としては4ビットフルアダー×3～12ビットフルアダーの機能回路がそれぞれ必要となる。これに対し、ウインドウをn個追加するのに必要な回路は、32ビット長レジスタ× $16 \times n$ であるので、第1図の構成のほうがコストパフォーマンスが非常に優れていることがわ

かる。

また、上記実施例ではハードウェアで8本/4本の切り替えを行なっているが、プロセッサ自身が切り替え命令を持つようにして、その命令を用いてソフトウェア上で切り替えを行なうようにしても良いことはもちろんである。

この場合、第1図に示すW-CCレジスタのような機能をはたすレジスタの内容を、所定の値に書き変える命令を、プロセッサの命令セットに設ければ良い。

#### 【発明の効果】

本発明によれば、複数のワーキングレジスタを有するシステムにおいて、ウインドウ(モジュール)に割り当てるレジスタ数を変化することができるため、システムの冗長性が無くなり、ハードウェア資源を無駄無く有効に利用することが可能となり、コストパフォーマンスの優れた、レジスタウインドウ方式によるマイクロプロセッサを得ることができるという可変方式の特長に加え、さらに、本発明の第1の発明によれば、レジスタ

ウインドウのレジスタの割り当て数の記憶を、ワーキングレジスタに対するアドレス情報に基づいて自動的に行なうようにしたので、固定方式の既存のプログラムをそのまま実行でき、かつウインドウのオーバーフロー／アンダーフローによるオーバーヘッドを低減することの可能なマイクロプロセッサを得ることができる。

また、本発明の第2の発明によれば、前記割り当て数の記憶を、CPUが実行できる命令の1つに基づいて行なうようにしたので、プログラム内のプロシージャの使用するレジスタの規模に応じて割り当て数の変更等をソフトウェアによって実現でき、従来技術による可変方式のプロセッサよりもさらにレジスタの使用効率を高めることができ、より使い勝手が良く、よりコストパフォーマンスの高い可変方式のウインドウによるマイクロプロセッサを提供することができる。

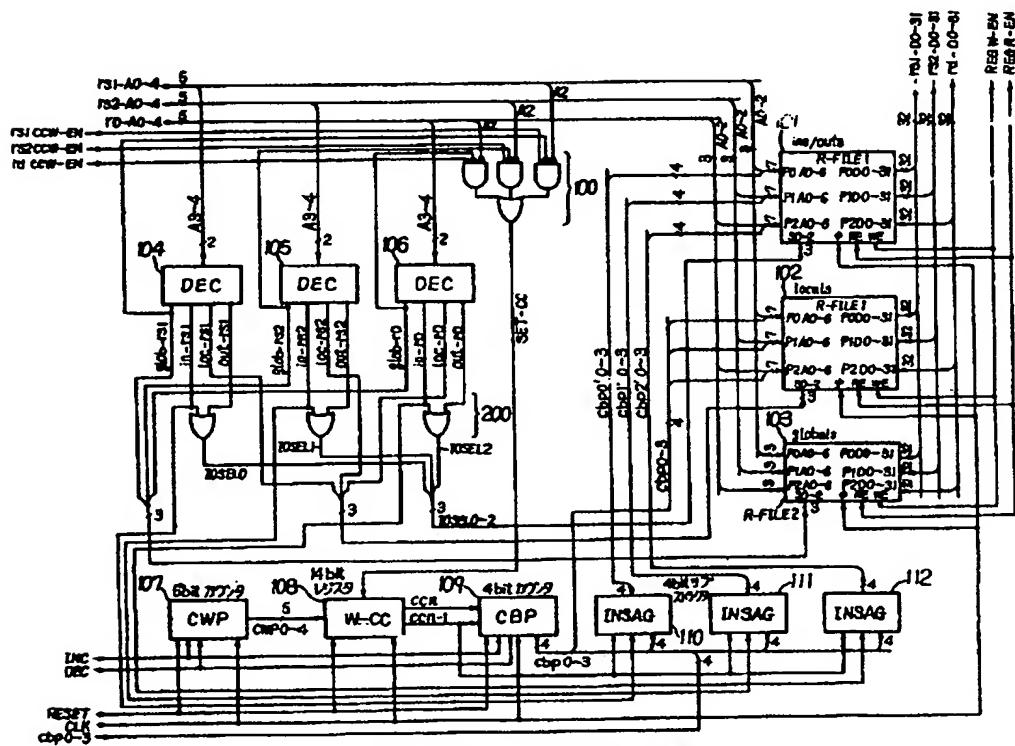
#### 4. 図面の簡単な説明

第1図は本発明の一実施例に係るマイクロプロ

セッサのブロック図、第2図は第1図のR-FILE1モジュール101の詳細例を示す説明図、第3図は第1図の装置の動作を説明するタイミングチャート、第4図はパイプライン構造の概略説明図である。

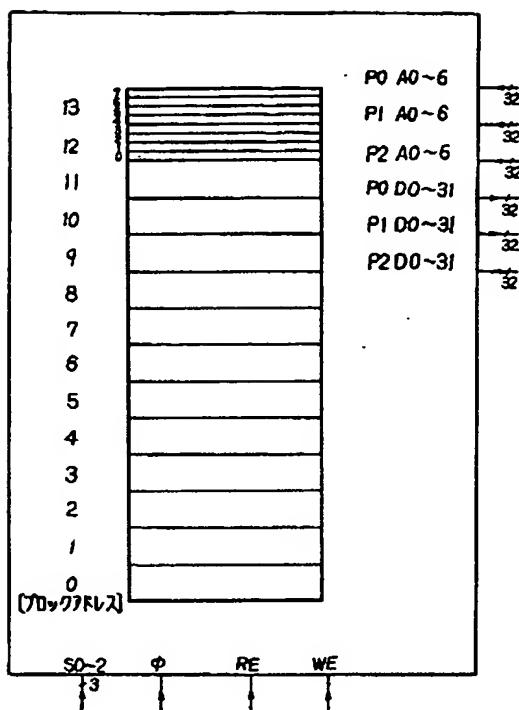
- 101…R-FILE1モジュール、
- 102…R-FILE2モジュール、
- 103…R-FILE2モジュール、
- 104, 105, 106…デコーダ(DEC)、
- 107…カレントウインドウポインタ(CWP)、
- 108…レジスタ(W-CC)、
- 109…カレントブロックポインタ(CBP)、
- 110, 111, 112…サブストラクタ(INSAG)。

出願人代理人 佐藤一雄



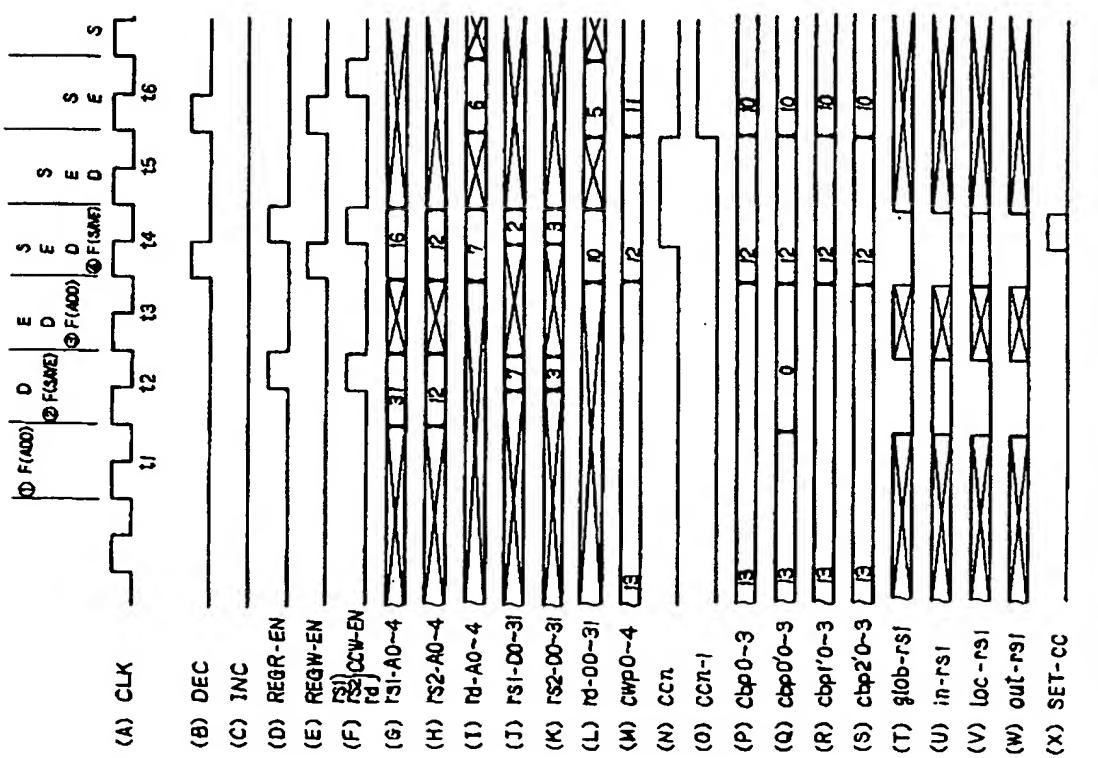
本発明の一実施例のブロック図

第1図

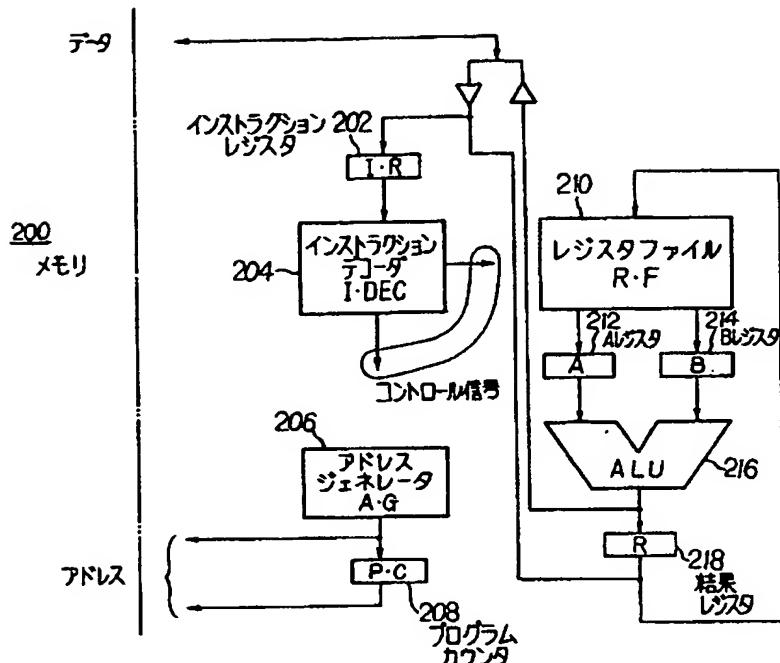


第1図のR-FILE 1モジュール101の内部構造説明図

第2図



第1図の構成の動作を説明するためのタイミングチャート  
第3図



パイプライン構造の概略説明図

第4図

## 手続補正書

平成 2 年 2 月 1 日

特許庁長官 吉田文毅

1 事件の表示 平成 1 年特許第 37805 号

2 発明の名称 マイクロプロセッサ

3 補正をする者 事件との関係 特許出願人  
(307) 株式会社 東芝 (ほか 1 名)

4 代理人 (都便番号 100)  
東京都千代田区丸の内三丁目2番3号  
〔電話東京 (211)2321 大代表〕  
6428 弁理士 佐藤一

5 補正により する請求項の数

6 補正の対象 明細書の「特許請求の範囲」及び「発明の詳細な説明」の項

## 8 補正の内容

(1) 特許請求の範囲を別紙の通り訂正する。

(2) 明細書第 7 頁下から 3 行目～第 8 頁下から 5 行目に記載の「本発明の第 2 …構成される。」を下記の通り補正する。

## 記

本願発明の第 2 のマイクロプロセッサは、複数個のレジスタウインドウを有し、前記各レジスタウインドウは複数個のレジスタを有するマイクロプロセッサであって、

前記各レジスタは前記レジスタウインドウのうちの複数のものに対して割り当てられており、さらには、

プログラム上の個々のプロシージャに対して前記レジスタウインドウのうちのいずれかを割り当てることにより、前記各プロシージャ上で使用されるワーキングレジスタを決定する第 1 決定手段と、

前記各レジスタウインドウへのレジスタ割り当て数を、実行できる命令の 1 つに基づいて記憶す

ト  
2.2.2  
正  
明  
細  
書

る記憶手段と、

前記記憶手段中の前記割り当て数に基づいて、前記各レジスタウィンドウが前記レジスタのうちのどれによって構成されるかを決定する第2決定手段と、を備えるものとして構成される。

特許請求の範囲

1. 複数個のレジスタウィンドウを有し、前記各レジスタウィンドウは複数個のレジスタを有するマイクロプロセッサであって、

前記各レジスタは前記レジスタウィンドウのうちの複数のものに対して割り当てられており、さらには、

プログラム上の個々のプロシージュに対して前記レジスタウィンドウのうちのいずれかを割り当てることにより、前記各プロシージュ上で使用されるワーキングレジスタを決定する第1決定手段と、

前記各レジスタウィンドウへのレジスタ割り当て数を、前記ワーキングレジスタに対するアドレス情報に基づいて自動的に記憶する記憶手段と、

前記記憶手段中の前記割り当て数に基づいて、前記各レジスタウィンドウが前記レジスタのうちのどれによって構成されるかを決定する第2決定手段と、を備えることを特徴とするマイクロプロセッサ。

2. 複数個のレジスタウィンドウを有し、前記各レジスタウィンドウは複数個のレジスタを有するマイクロプロセッサであって、

前記各レジスタは前記レジスタウィンドウのうちの複数のものに対して割り当てられており、さらには、

プログラム上の個々のプロシージュに対して前記レジスタウィンドウのうちのいずれかを割り当てることにより、前記各プロシージュ上で使用されるワーキングレジスタを決定する第1決定手段と、

前記各レジスタウィンドウへのレジスタ割り当て数を、実行できる命令の1つに基づいて記憶する記憶手段と、

前記記憶手段中の前記割り当て数に基づいて、前記各レジスタウィンドウが前記レジスタのうちのどれによって構成されるかを決定する第2決定手段と、を備えることを特徴とするマイクロプロセッサ。

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**